NAME: _____

CSE 331

Introduction to Algorithm Analysis and Design

Sample Final Exam: Fall 2011

Atri Rudra

DIRECTIONS:

- Closed Book, Closed Notes except for two $8\frac{1}{2}$" $\times 11$" review sheet.
- Time Limit: 2 hours 30 minutes.
- Answer the problems on the exam paper.
- Each problem starts on a new page.
- Make sure you write your NAME on the paper.
- If you need extra space use the back of a page.
- Problem 5b is a bonus problem.

1	/10
2	/30
3	/25
4	/20
5a	/15
5b	/10
Total	/100
Bonus	/10

FEW GENTLE REMINDERS:

- You can quote any result that we covered in class or any problem that was there in a homework (but remember to explicitly state where you are quoting a result from).
- If you get stuck on some problem for a long time, move on to the next one.
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You should be better off by **first reading all questions** and answering them in the order of what you think is the easiest to the hardest problem. Keep the points distribution in mind when deciding how much time to spend on each problem.
- Spend time on the bonus problem only if you are done with the rest of the exam.
- Finally, relax and enjoy the exam! (If not think of time when you'll be done with 331!)

1. ($5 \times 2 = 10$ points) Answer True or False to the following questions. **No justification** is required. (Recall that a statement is true only if it is logically true in all cases while it is is false if it is not true in some case).

   (a) Depth First Search (DFS) is a linear time algorithm.

   (b) n is $O\left((\log n)^{\log n}\right)$.

   (c) There is no algorithm that can compute the Minimum Spanning Tree (MST) of a graph on n vertices and m edges in time asymptotically faster than $O(m \log n)$.

   (d) Let G be a graph with a negative cycle. Then there is no pair of vertices that has a finite cost shortest path.

   (e) Every computational problem on input size n can be solved by an algorithm with running time polynomial in n.

2. ($5 \times 6 = 30$ points) Answer True or False to the following questions and **briefly JUSTIFY** each answer. A correct answer with no or totally incorrect justification will get you 2 out of the total 6 points. (Recall that a statement is true only if it is logically true in all cases while it is is false if it is not true in some case).

   (a) The following algorithm to check if the input number n is a prime number runs in polynomial time.

   > For every integer $2 \leq i \leq \sqrt{n}$, check if i divides n. If so declare n to be *not* a prime. If no such i exists, declare n to be a prime.

   (b) Let $G = (V, E)$ be a directed graph with positive costs, i.e. $c_e \geq 0$ for every $e \in E$. The following is true for every real number $\delta > 0$. Consider the instance where the input graph is still G but the costs are now $c'_e = c_e + \delta$ (for every $e \in E$). Then for some distinct $s, t \in V$, the shortest $s - t$ path in the two instances are different.

   (c) Every weighted graph has a unique Minimum Spanning Tree (MST).

(d) The weighted interval scheduling problem on n jobs can be solved in $O(n)$ time.

(e) Given n integers $a_1, \ldots, a_n$, the third smallest number among $a_1, \ldots, a_n$ can be computed in $O(n)$ time.

3. ($10 + 15 = 25$ points) In this problem you will show that the naive recursive algorithm (that we saw in class) to compute the value of the optimal schedule for the weighted interval scheduling problem takes exponential time.

   (a) As a warmup, we will begin with a recurrence relation, that does not have anything to do with the weighted interval scheduling per se. Consider the following recurrence relation: $F(0) = 0$, $F(1) = 1$ and for every $n \geq 2$, $F(n) = F(n-1) + F(n-2)$. Prove that $F(n) \geq \left(\frac{3}{2}\right)^{n-2}$. (Note that you have to prove a *lower bound*.)

   Hint: Use the guess and verify using induction technique of solving recurrences.

(b) Using the part (a) or otherwise, prove that for every $n \geq 2$, the following recursive algorithm for the weighted interval scheduling problem takes $\Omega((1.5)^n)$ time. (Recall that the input to the weighted interval scheduling problem are n jobs where the ith job is the tuple (s_i, f_i, v_i). Further, note that we assume that $f_1 \leq f_2 \leq \cdots \leq f_n$ and the values $p(j)$ are known. Finally, recall that $p(j)$ is the the largest job index $i \leq j$ such that $s_j \geq f_{p(j)}$.)

Compute-Opt(j)
If $j == 0$ **return** 0.
return $\max\{v_j + \textbf{Compute-Opt}(p(j)), \textbf{Compute-Opt}(j-1)\}$.

4. ($5 + 15 = 20$ points) A *boolean* polynomial $P(X)$ of *degree* d is the formal polynomial $P(X) = \sum_{i=0}^{d} p_i X^i$, where for every $0 \leq i \leq d$, $p_i \in \{0, 1\}$ (also called the ith *coefficient*) and X is a variable. Note that a polynomial is specified once the coefficients $p_0, \ldots, p_d$ are specified. (E.g. $X^4 + X^2 + X + 1$ is a degree 4 polynomial).

   Given two polynomials $P(X)$ and $Q(X)$ of degree at most $n-1$, their product $R(X) = P(X) \cdot Q(X)$ is defined as the formal polynomial

$$\left(\sum_{i=0}^{n-1} p_i X^i\right)\left(\sum_{i=0}^{n-1} q_i X^i\right) = \sum_{i=0}^{2n-2}\left(\sum_{j=\max(0,i-n+1)}^{i} p_j q_{i-j}\right) X^i.$$

   For example, if $P(X) = X^2 + 1$ and $Q(X) = X^2 + X + 1$, then $P(X) \cdot Q(X) = X^4 + X^3 + 2X^2 + X + 1$. (Note that the resulting polynomial can have coefficients taking values outside of $\{0, 1\}$.)

   (a) Let $P(X) = X^3 + X + 1$ and $Q(X) = X^2 + X$. What is $P(X) \cdot Q(X)$?

(b) In this part, you will design an algorithm to solve the polynomial multiplication problem. In particular, the input to the problem for input size $n \geq 1$ are the coefficients of the two polynomials $p_0, \ldots, p_{n-1}$ and $q_0, \ldots, q_{n-1}$. The output should be the coefficient $r_0, \ldots, r_{2n-2}$, where $P(X) = \sum_{i=0}^{n-1} p_i^i X^i$, $Q(X) = \sum_{i=0}^{n-1} q_i X^i$, $R(X) = \sum_{i=0}^{2n-2} r_i X^i$ such that $R(X) = P(X) \cdot Q(X)$. It might be useful to know that the polynomial addition and substraction have natural definitions: $P(X) + Q(X) = \sum_{i=0}^{n-1}(p_i + q_i)X^i$ and $P(X) - Q(X) = \sum_{i=0}^{n-1}(p_i - q_i)X^i$.

Design a divide and conquer algorithm that runs in time asymptotically faster than $O(n^2)$. Justify the correctness of your algorithm (formal proof is not required). Also state the running time of your algorithm (and very briefly justify it.)

Hint: Think of a similar problem we solved in class by a divide and conquer algorithm.

5. You are almost done!

   (a) (15 points) You given as input n real numbers $x_1, \ldots, x_n$. Design an efficient algorithm that uses the minimum number m of unit intervals $[\ell_i, \ell_i+1)$ $(1 \leq i \leq m)$ that cover all the input numbers. A number x_j is covered by an interval $[\ell_i, \ell_i+1)$ if $\ell_i \leq x_j < \ell_i + 1$. Argue why your algorithm is correct (formal proof is not required).

   For example, consider the input for $n = 4$: $0.1, 0.9, 1.1, 1.555$. Then the two intervals $[0.1, 1.1)$ and $[1.1, 2.1)$ cover all the input numbers (i.e. in this case $m = 2$).

(b) (**Bonus**) (10 points) I am too lazy to put in a bonus problem here but there will be one in the actual final exam.